Spring Cloud로 개발하는 마이크로서비스 애플리케이션

목차

Part II

Section 14.

Microservice 모니터링

- Hystrix Dashboard + Turbine Server
- Micrometer
- Prometheus
- Grafana

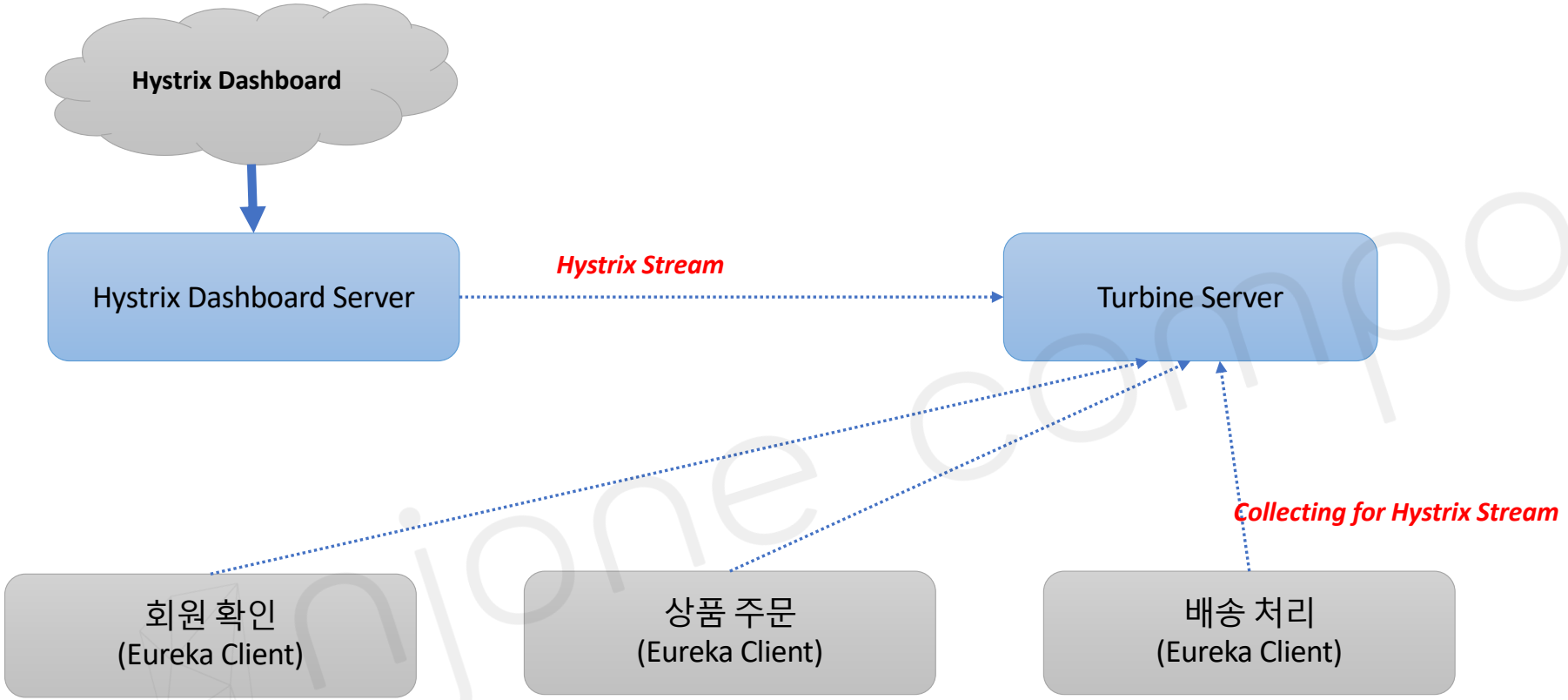
Turbine Server

- 마이크로서비스에 설치된 Hystrix 클라이언트의 스트림을 통합
  - 마이크로서비스에서 생성되는 Hystrix 클라이언트 스트림 메시지를 터빈 서버로 수집

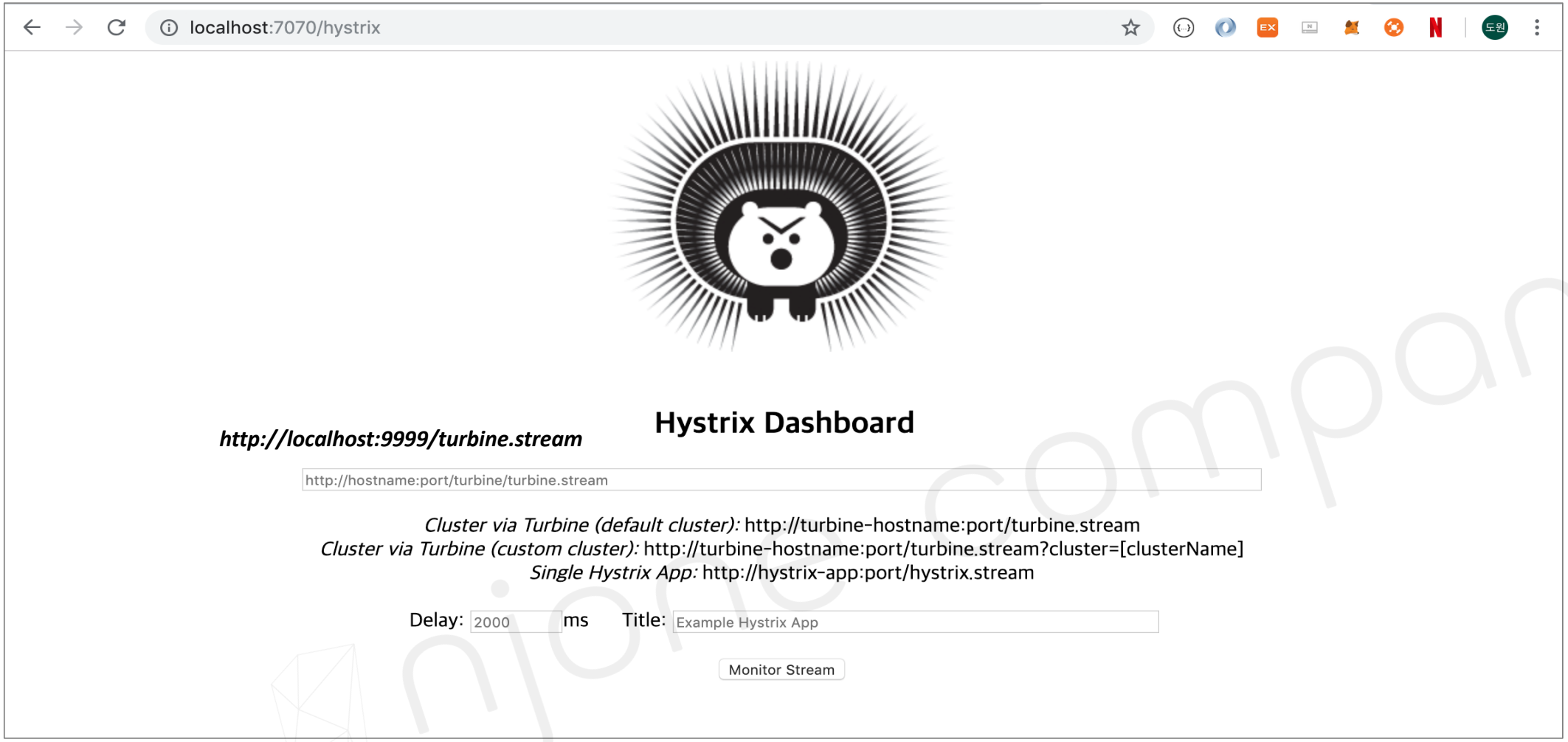
```
#Turbine Server
turbine:
 appConfig:
            msa-service-product-order,
            msa-service-product-member,
            msa-service-product-status
 clusterNameExpression: new String("default")
```

Hystrix Dashboard

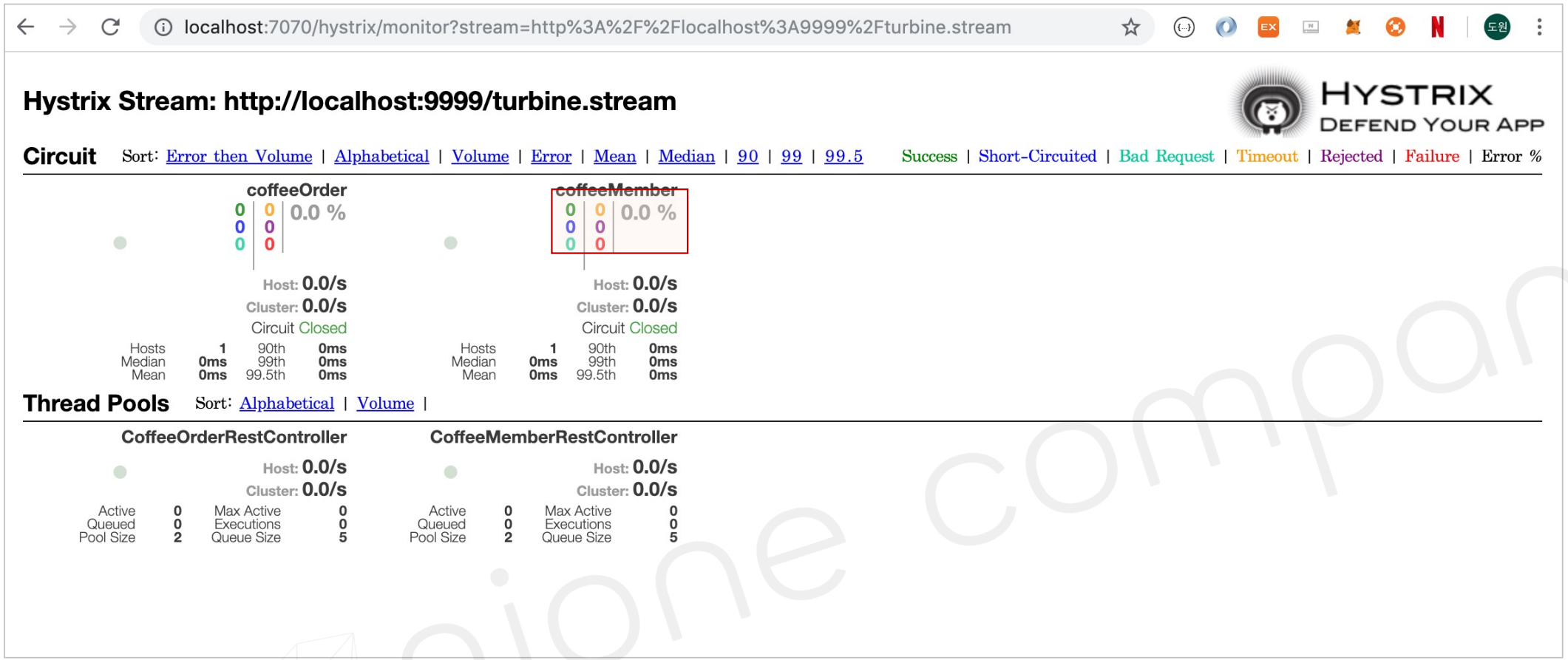
- Hystrix 클라이언트에서 생성하는 스트림을 시각화
  - Web Dashboard



Hystrix Dashboard



Hystrix Dashboard



Micrometer + Monitoring

CURRENT	REPLACEMENT
Hystrix	Resilience4j
Hystrix Dashboard / Turbine	Micrometer + Monitoring System
Ribbon	Spring Cloud Loadbalancer
Zuul 1	Spring Cloud Gateway
Archaius 1	Spring Boot external config + Spring Cloud Config

Replacements technologies. Taken from [1]

Micrometer

- Micrometer
  - https://micrometer.io/
  - JVM기반의 애플리케이션의 Metrics 제공
  - Spring Framework 5, Spring Boot 2부터 Spring의 Metrics 처리
  - Prometheus등의 다양한 모니터링 시스템 지원

- Timer
  - 짧은 지연 시간, 이벤트의 사용 빈도를 측정
  - 시계열로 이벤트의 시간, 호출 빈도 등을 제공
  - @Timed 제공

Microservice 수정

```xml
<!-- micrometer -->
<dependency>
    <groupId>io.micrometer</groupId>
    <artifactId>micrometer-registry-prometheus</artifactId>
</dependency>
```

pom.xml

```yaml
management:
  endpoints:
    web:
      exposure:
        include: refresh,health,beans,busrefresh,info,prometheus,metrics
```

application.yml (Users microservice)

Microservice 수정

```java
@GetMapping("/health_check")
@Timed(value="users.status", longTask = true)
public String status(HttpServletRequest request) {
    return String.format("It's Working in User Service"
             + ", port(local.server.port)=" + env.getProperty("local.server.port")
            + ", port(server.port)=" + env.getProperty("server.port")
            + ", with token secret=" + env.getProperty("token.secret")
            + ", with token time=" + env.getProperty("token.expiration_time"));
}

@GetMapping("/welcome")
@Timed(value="users.welcome", longTask = true)
public String welcome() {
      return env.getProperty("greeting.message");
    log.info(">>>GET");
    return greeting.getMessage();
}
```

127.0.0.1:8000/user-service/actuator/metrics

```
"tomcat.sessions.active.max",
"tomcat.sessions.alive.max",
"tomcat.sessions.created",
"tomcat.sessions.expired",
"tomcat.sessions.rejected",
"users.status",
"users.welcome",
"zipkin.reporter.messages",
```

Microservice 수정

- metrics 정보

주의 요함 | 10.204.136.204:52257/actuator/prometheus

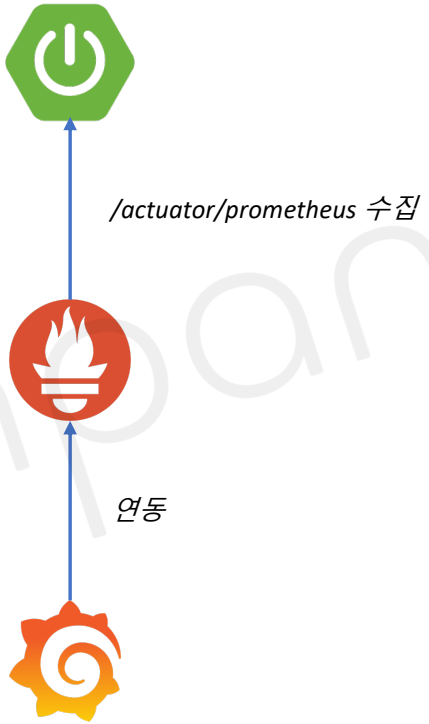
```
# HELP logback_events_total Number of error level events that made it to the logs
# TYPE logback_events_total counter
logback_events_total{level="warn",} 2.0
logback_events_total{level="debug",} 0.0
logback_events_total{level="error",} 0.0
logback_events_total{level="trace",} 0.0
logback_events_total{level="info",} 25.0
# HELP process_files_max_files The maximum file descriptor count
# TYPE process_files_max_files gauge
process_files_max_files 10240.0
# HELP jvm_gc_pause_seconds Time spent in GC pause

# HELP users_welcome_seconds_max
# TYPE users_welcome_seconds_max gauge
users_welcome_seconds_max{method="GET",uri="/welcome",} 0.0
# HELP users_welcome_seconds
# TYPE users_welcome_seconds summary
users_welcome_seconds_active_count{method="GET",uri="/welcome",} 0.0
users_welcome_seconds_duration_sum{method="GET",uri="/welcome",} 0.0

# HELP users_status_seconds_max
# TYPE users_status_seconds_max gauge
users_status_seconds_max{method="GET",uri="/health_check",} 0.0
# HELP users_status_seconds
# TYPE users_status_seconds summary
users_status_seconds_active_count{method="GET",uri="/health_check",} 0.0
users_status_seconds_duration_sum{method="GET",uri="/health_check",} 0.0
```

Prometheus + Grafana

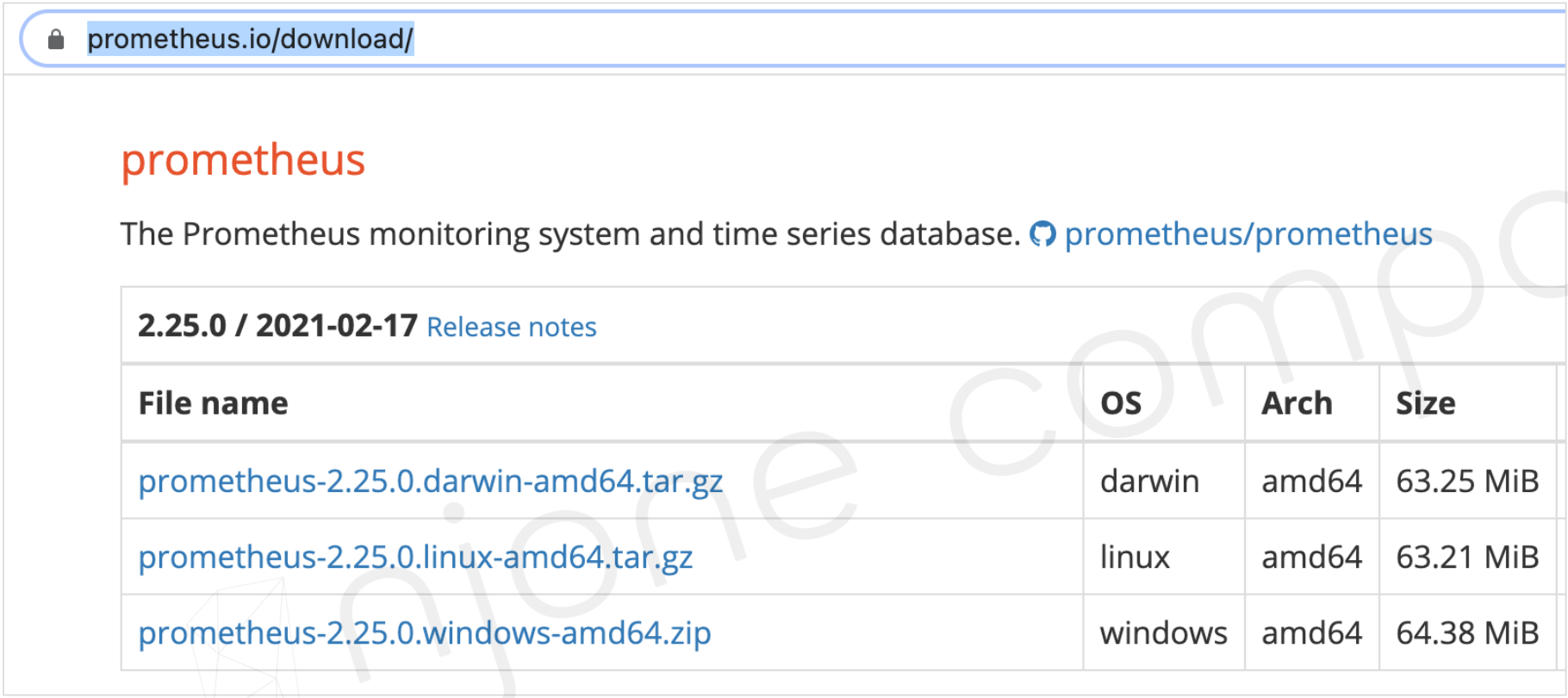
- Prometheus
  - Metrics를 수집하고 모니터링 및 알람에 사용되는 오픈소스 애플리케이션
  - 2016년부터 CNCF에서 관리되는 2번째 공식 프로젝트
    - Level DB → Time Series Database(TSDB)
  - Pull 방식의 구조와 다양한 Metric Exporter 제공
  - 시계열 DB에 Metrics 저장 → 조회 가능 (Query)



- Grafana
  - 데이터 시각화, 모니터링 및 분석을 위한 오픈소스 애플리케이션
  - 시계열 데이터를 시각화하기 위한 대시보드 제공

Prometheus 설치

- Prometheus 다운로드
  - https://prometheus.io/download/



Prometheus

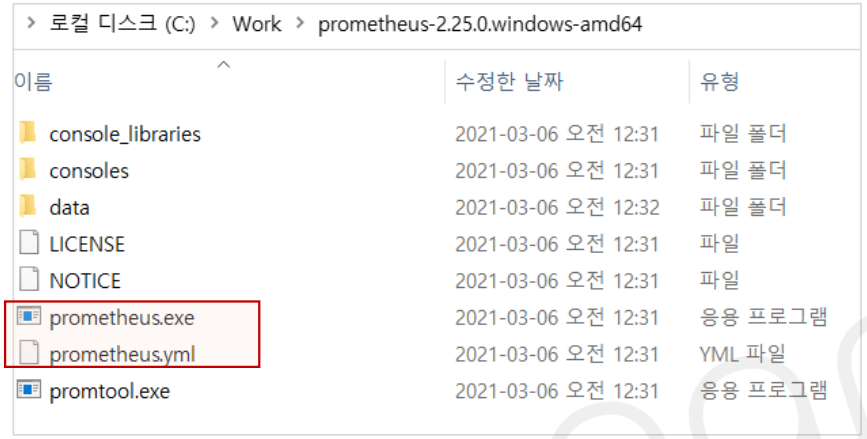
- prometheus.yml 파일 수정
  - target 지정

```
21  scrape_configs:
22    # The job name is added as a label `job=<job_name>` to any timeseries
      scraped from this config.
23    - job_name: 'prometheus'
24
25      # metrics_path defaults to '/metrics'
26      # scheme defaults to 'http'.
27
28      static_configs:
29      - targets: ['localhost:9090']
30
31    - job_name: 'user-service'
32      scrape_interval: 15s
33      metrics_path: '/user-service/actuator/prometheus'
34      static_configs:
35      - targets: ['localhost:8000']
```

Prometheus

- Prometheus 서버 실행

```
ls -l
total 334208
-rw-r--r--@  1 dowonlee  staff     11357  2 18 01:11 LICENSE
-rw-r--r--@  1 dowonlee  staff      3420  2 18 01:11 NOTICE
drwxr-xr-x@  4 dowonlee  staff       128  2 18 01:11 console_libraries
drwxr-xr-x@  9 dowonlee  staff       288  2 18 01:11 consoles
drwxr-xr-x  12 dowonlee  staff       384  3  5 00:00 data
-rwxr-xr-x@  1 dowonlee  staff  90568240  2 17 23:28 prometheus
-rw-r--r--@  1 dowonlee  staff      1089  3  1 23:28 prometheus.yml
-rwxr-xr-x@  1 dowonlee  staff  80523216  2 17 23:30 promtool
```

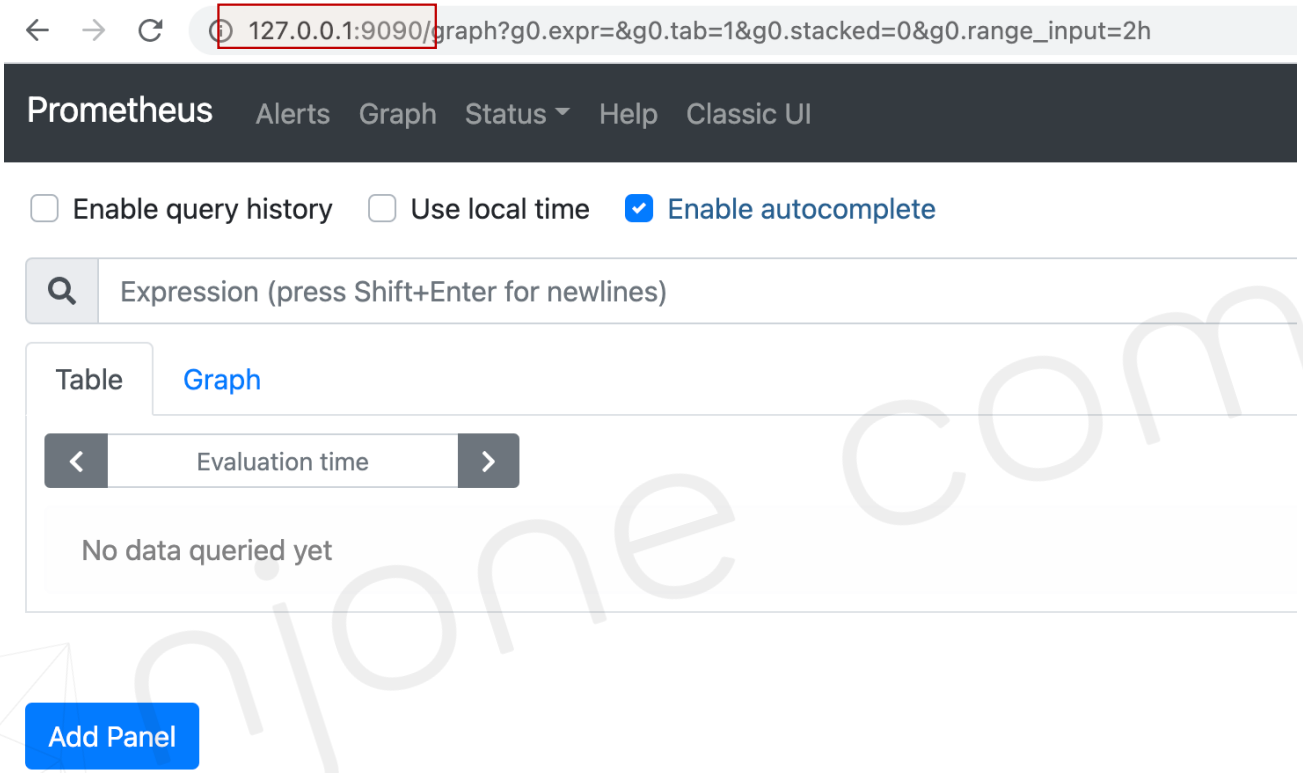


$ ./prometheus --config.file=prometheus.yml

```
./prometheus --config.file=prometheus.yml
level=info ts=2021-03-04T07:51:29.289Z caller=main.go:366 msg="No time or size retention was set so using the default time retentior
level=info ts=2021-03-04T07:51:29.289Z caller=main.go:404 msg="Starting Prometheus" version="(version=2.25.0, branch=HEAD, revision=
0e4bd0bd4c00ad6e2)"
level=info ts=2021-03-04T07:51:29.289Z caller=main.go:409 build_context="(go=go1.15.8, user=root@615f028225c9, date=20210217-14:26:1
level=info ts=2021-03-04T07:51:29.289Z caller=main.go:410 host_details=(darwin)
level=info ts=2021-03-04T07:51:29.289Z caller=main.go:411 fd_limits="(soft=256, hard=unlimited)"
level=info ts=2021-03-04T07:51:29.289Z caller=main.go:412 vm_limits="(soft=unlimited, hard=unlimited)"
level=info ts=2021-03-04T07:51:29.293Z caller=web.go:532 component=web msg="Start listening for connections" address=0.0.0.0:9090
level=info ts=2021-03-04T07:51:29.293Z caller=main.go:779 msg="Starting TSDB ..."
level=info ts=2021-03-04T07:51:29.295Z caller=tls_config.go:191 component=web msg="TLS is disabled." http2=false
```

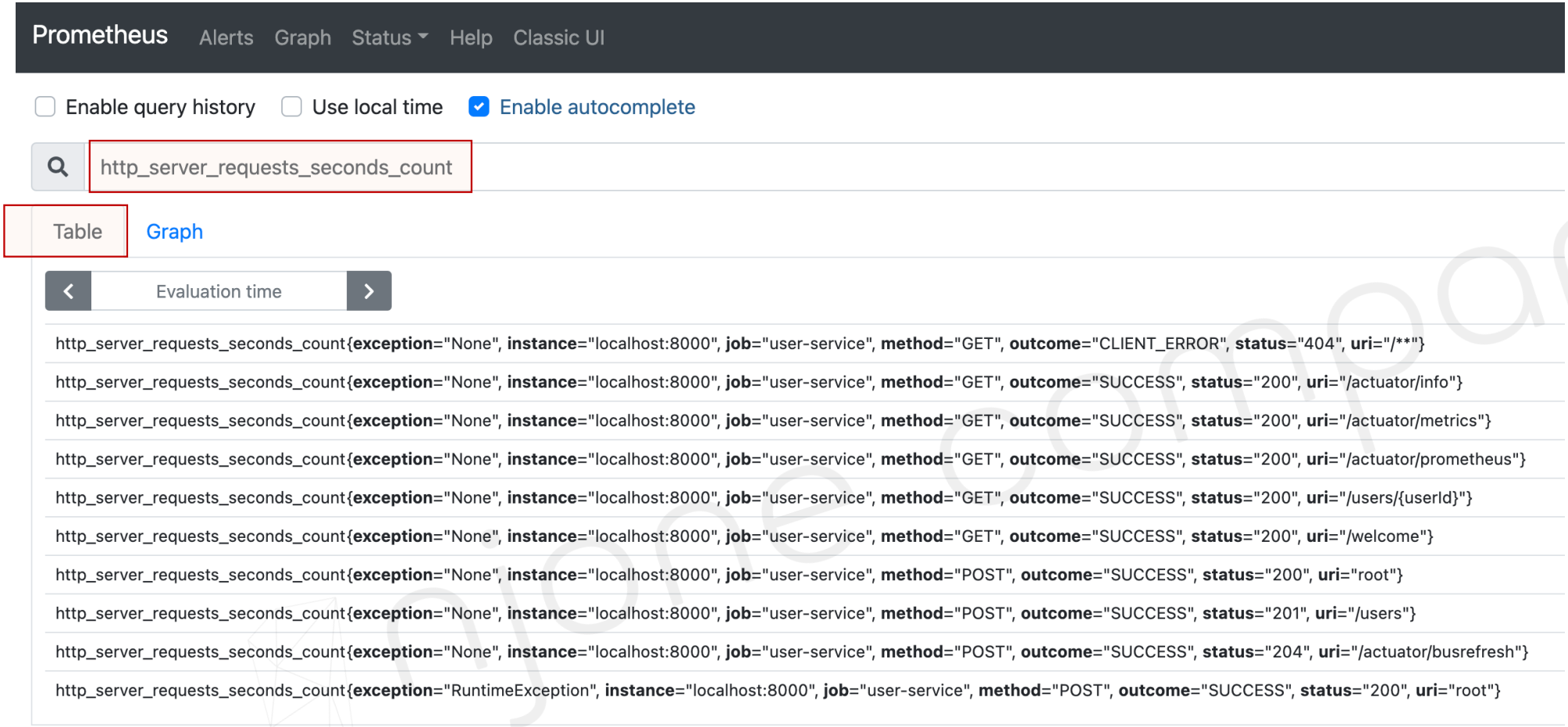
Prometheus

- Prometheus Dashboard
  - http://127.0.0.1:9090



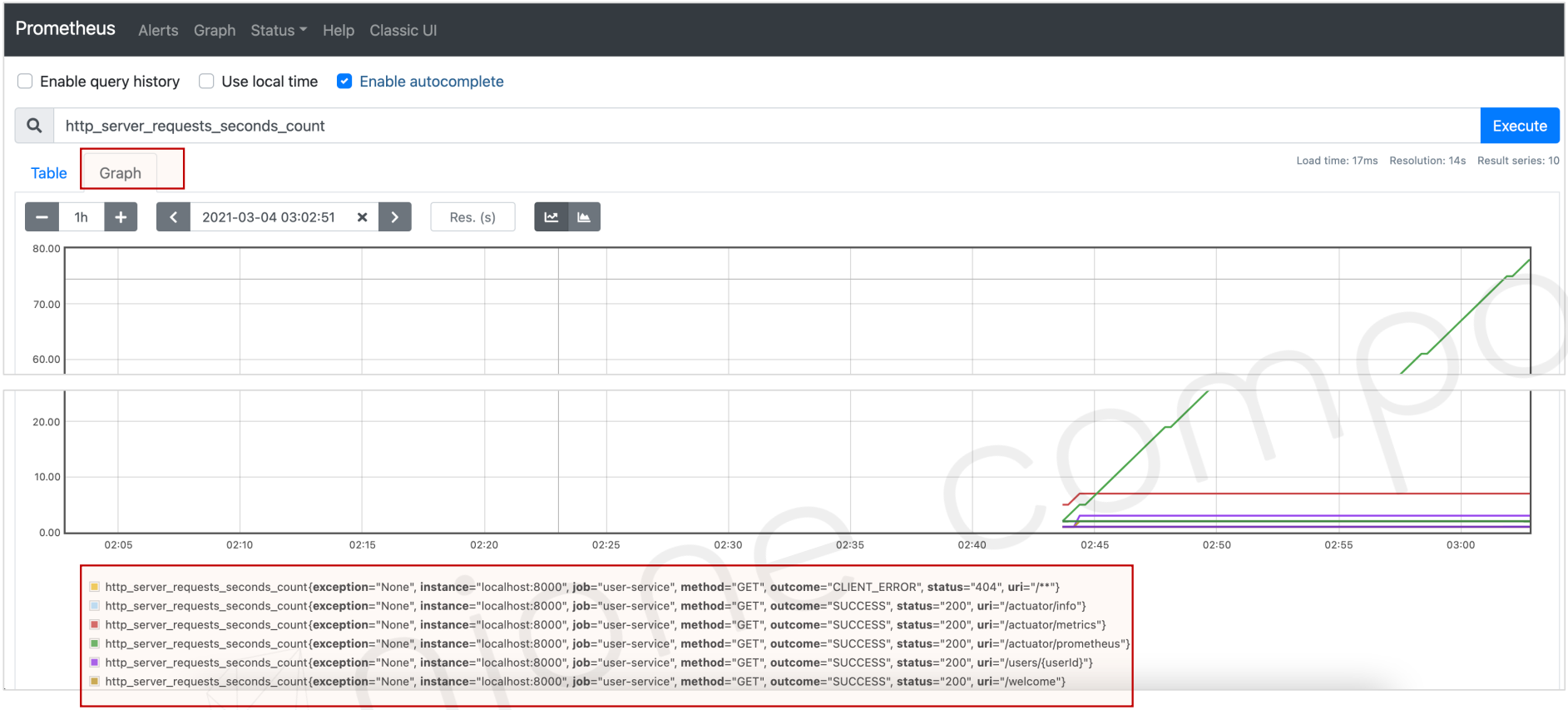
Prometheus

- metrics 검사 – Table



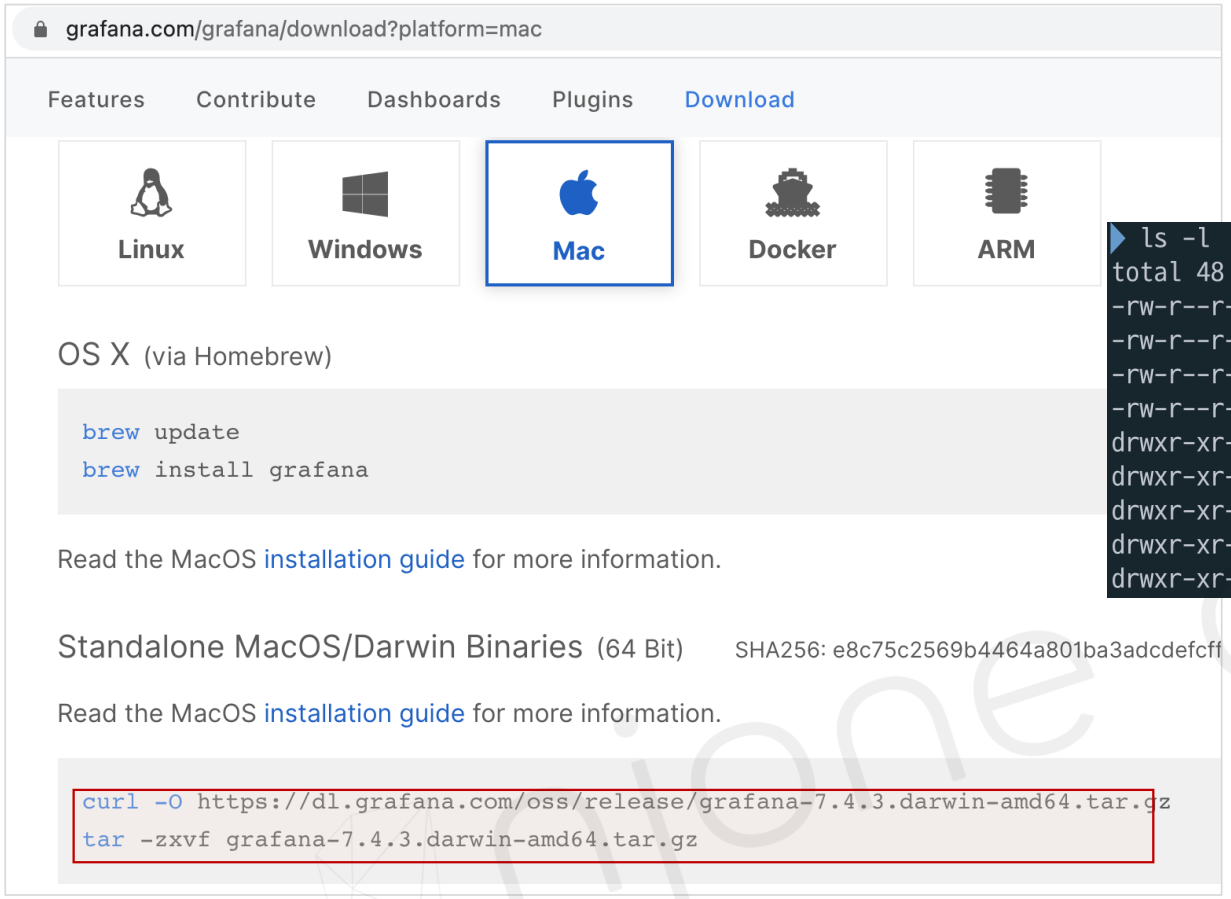
Prometheus

- metrics 검사 – Graph



Grafana

- Grafana 다운로드 – MacOS

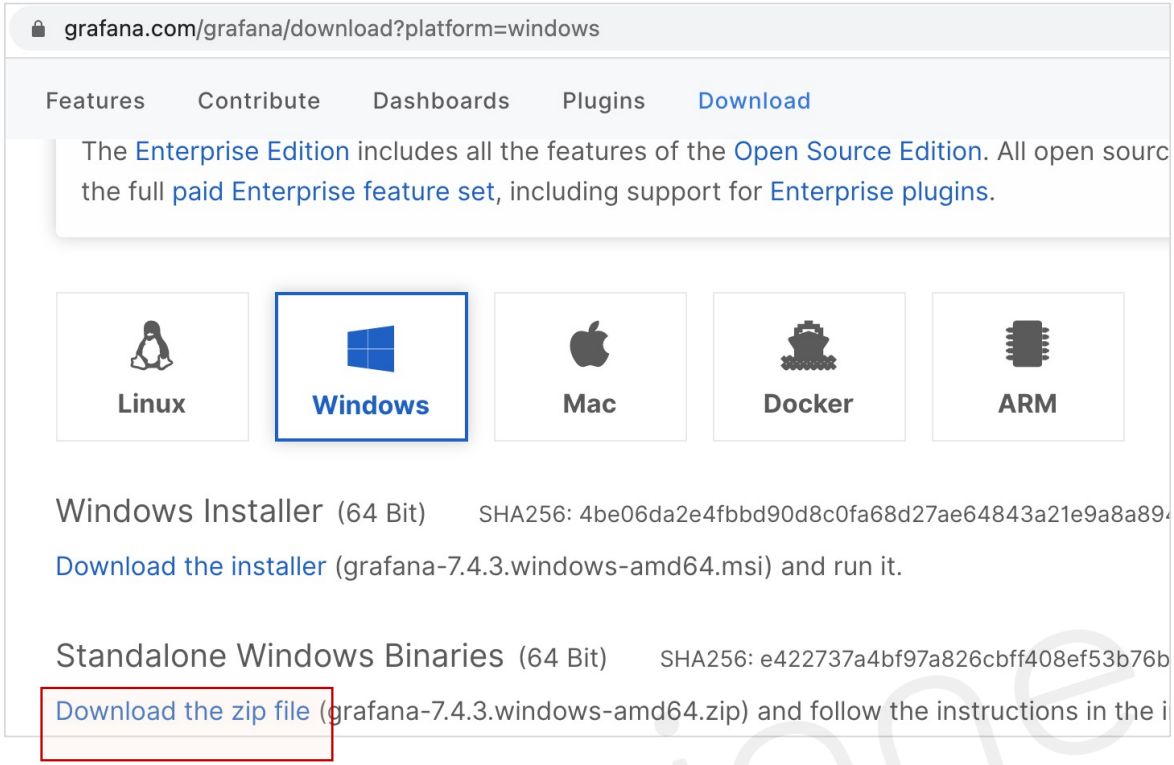


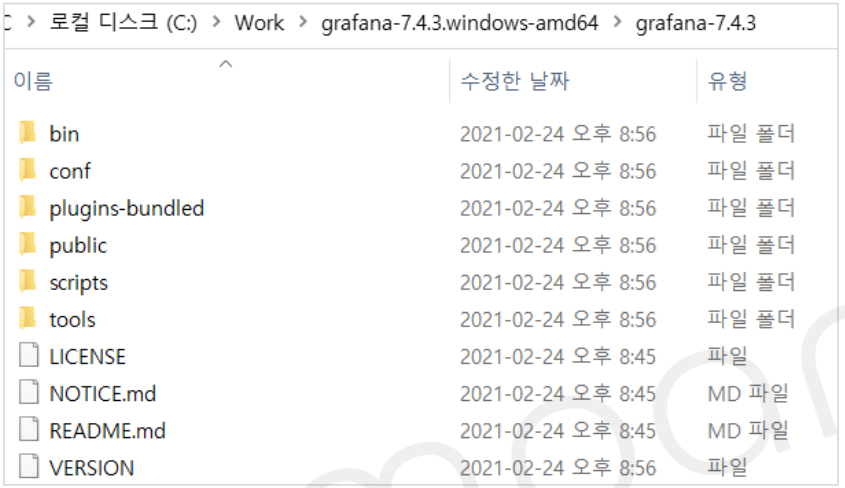

```
ls -l
total 48
-rw-r--r--@  1 dowonlee  staff  11343  2 24 20:45 LICENSE
-rw-r--r--@  1 dowonlee  staff    108  2 24 20:45 NOTICE.md
-rw-r--r--@  1 dowonlee  staff   2804  2 24 20:45 README.md
-rw-r--r--@  1 dowonlee  staff      5  2 24 20:56 VERSION
drwxr-xr-x@  6 dowonlee  staff    192  2 24 20:56 bin
drwxr-xr-x@  7 dowonlee  staff    224  2 24 20:56 conf
drwxr-xr-x@  3 dowonlee  staff     96  2 24 20:56 plugins-bundled
drwxr-xr-x@ 13 dowonlee  staff    416  2 24 20:56 public
drwxr-xr-x@ 29 dowonlee  staff    928  2 24 20:56 scripts
```

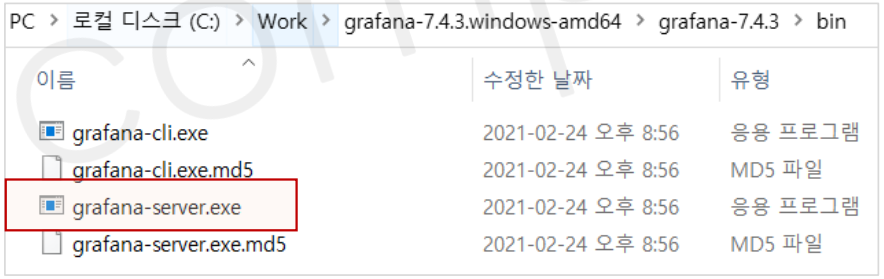
$ ./bin/grafana-server web

Grafana

- Grafana 다운로드 – Windows



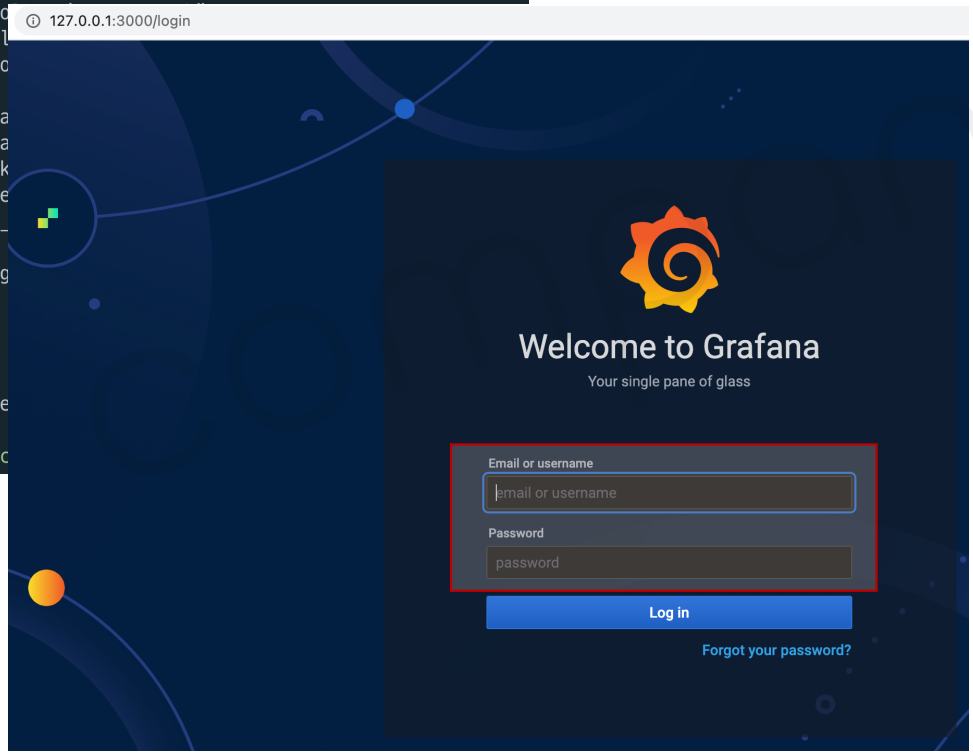




Grafana

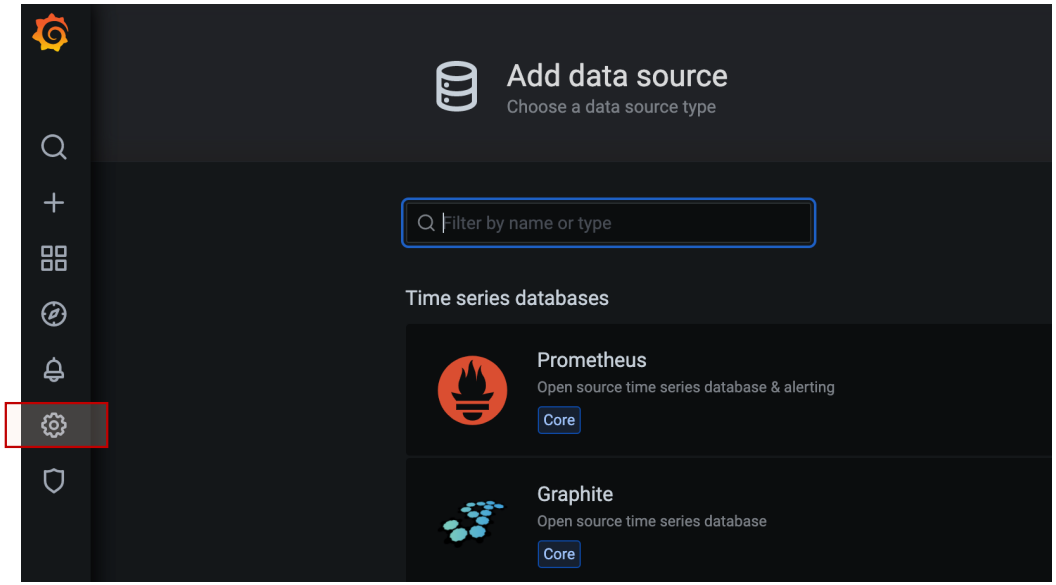
- Grafana 실행
  - http://127.0.0.1:3000
  - ID: admin, PW: admin

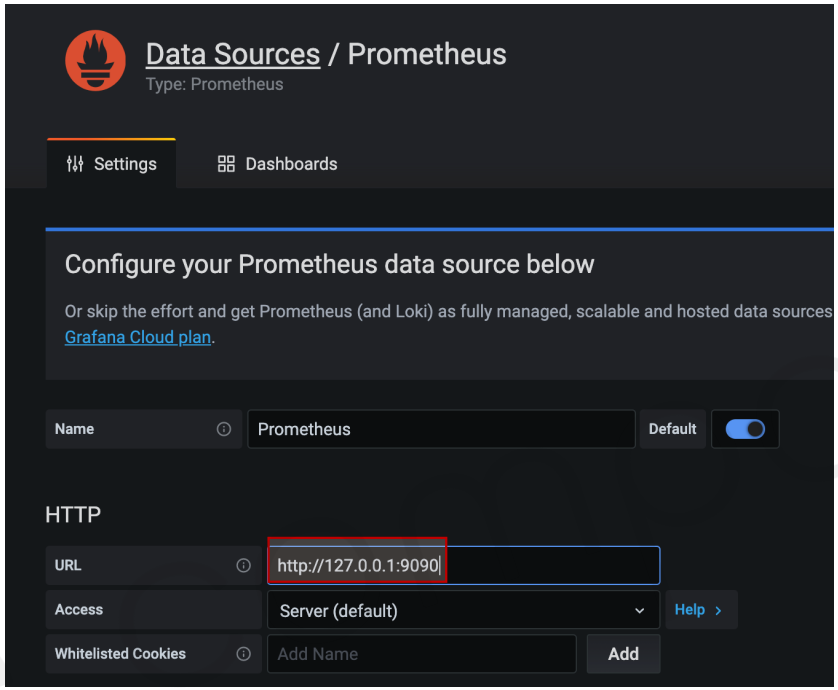
```
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="Add OAuth expiry to user_auth"
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="Add index to user_id co
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="create server_lock tabl
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="add index server_lock.o
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="create user auth token
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="add unique index user_a
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="add unique index user_a
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="add index user_auth_tok
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="create cache_data table
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="add unique index cache_
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="create short_url table
INFO[03-06|00:57:08] Executing migration                  logger=migrator id="add index short_url.org
INFO[03-06|00:57:08] Created default admin                logger=sqlstore user=admin
INFO[03-06|00:57:08] Created default organization         logger=sqlstore
INFO[03-06|00:57:08] Starting plugin search               logger=plugins
INFO[03-06|00:57:08] Registering plugin                   logger=plugins id=input
INFO[03-06|00:57:08] External plugins directory created   logger=plugins directory=/Users/dowonlee/De
ns
INFO[03-06|00:57:08] HTTP Server Listen                   logger=http.server address=[::]:3000 protoc
```

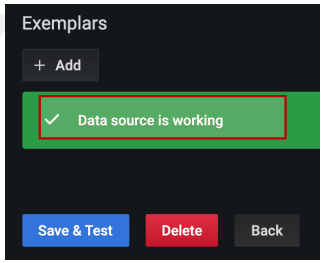


Grafana

- Grafana – Prometheus 연동

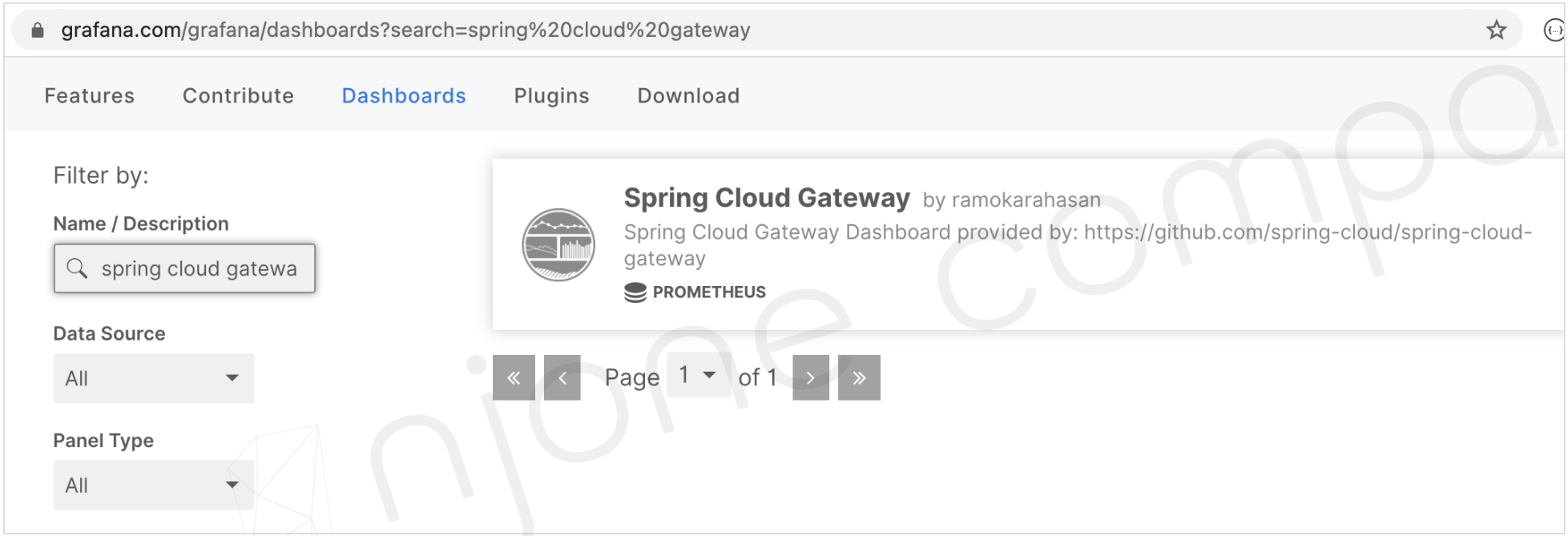






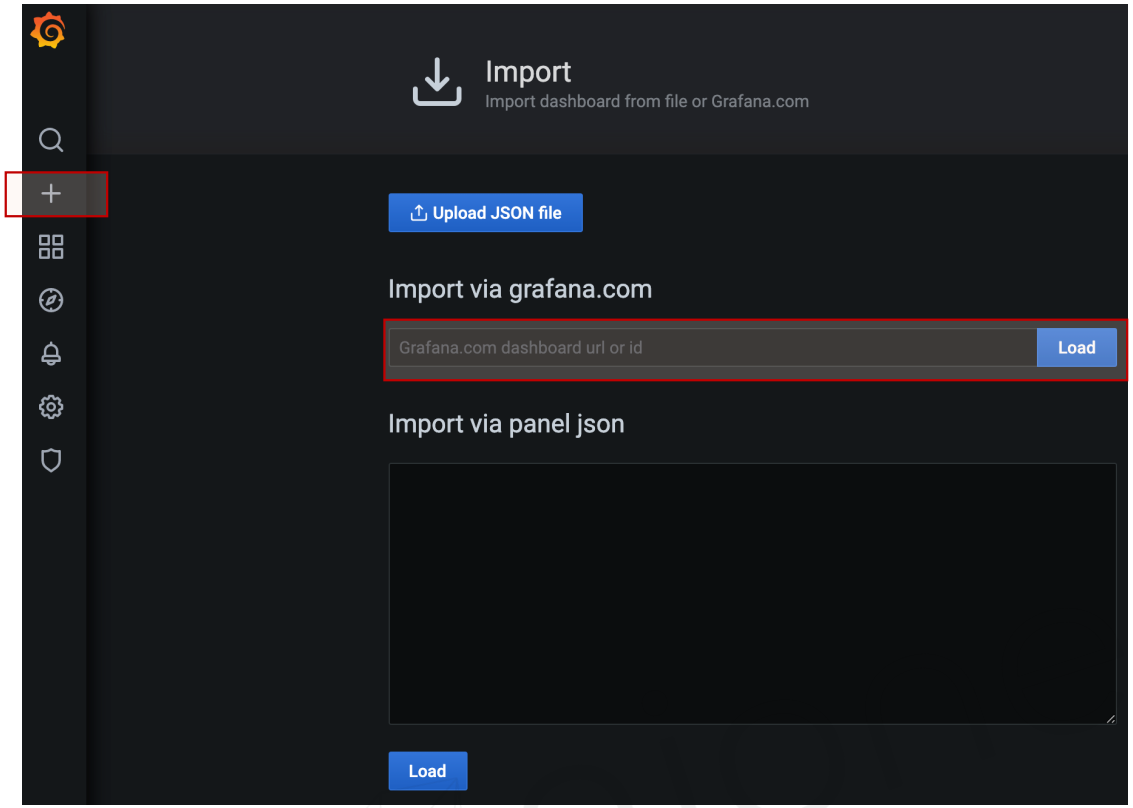
Grafana

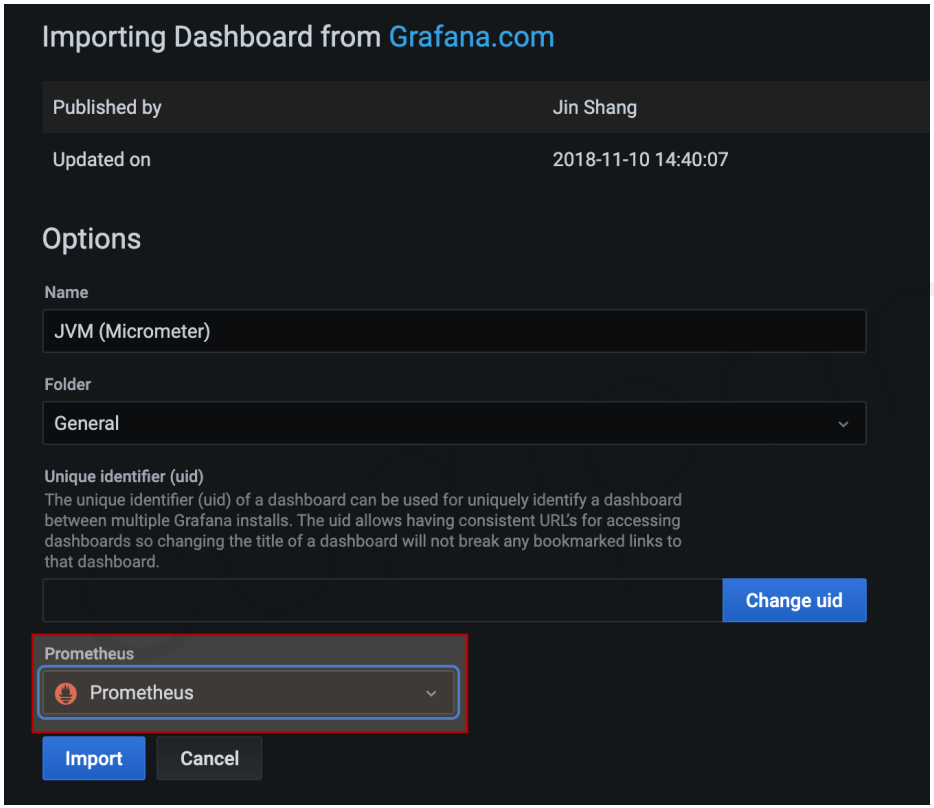
- Grafana Dashboard
  - JVM(Micrometer)
  - Prometheus
  - Spring Cloud Gateway



Grafana

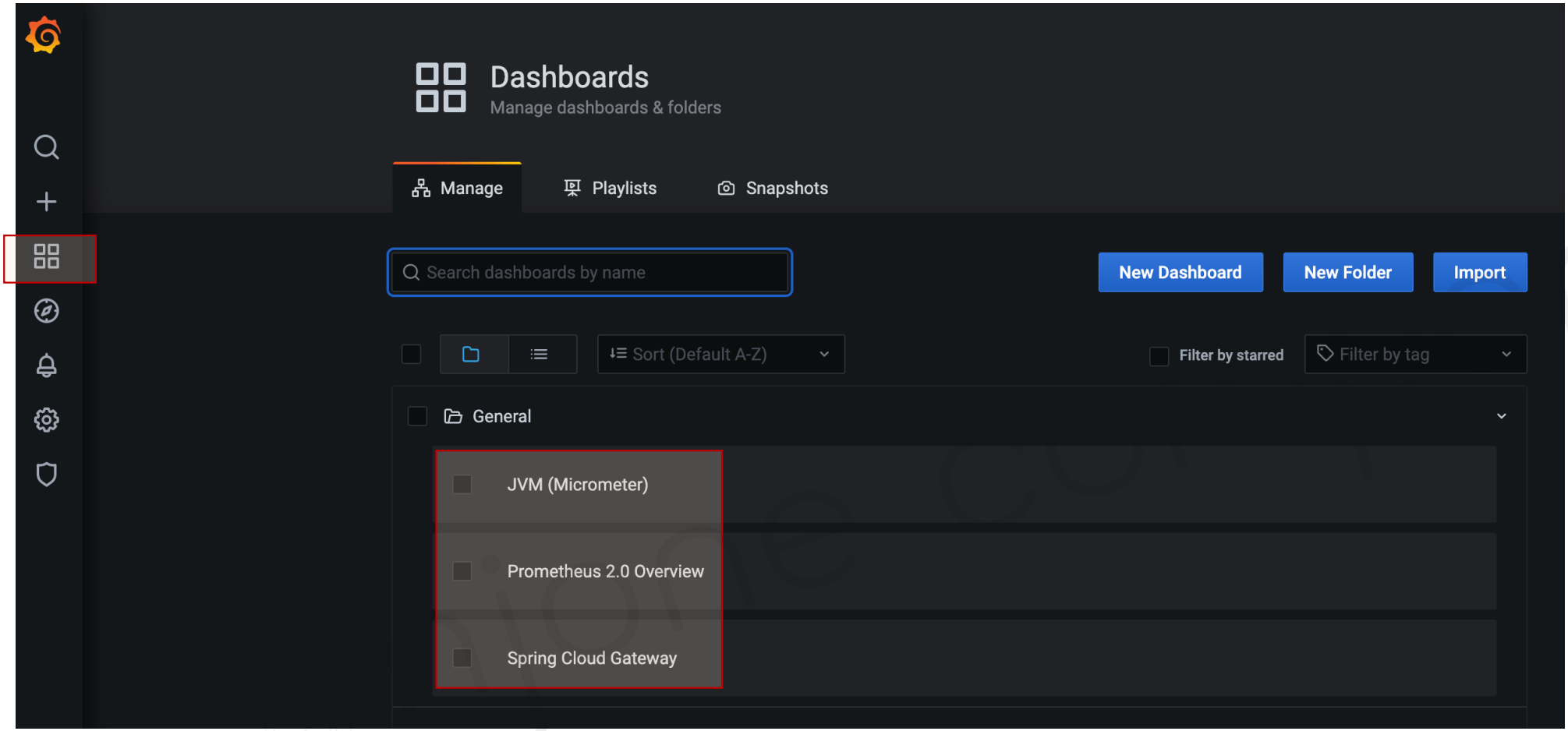
- Grafana – Prometheus 연동





Grafana

- Grafana Dashboard



Grafana

- ***Spring Cloud Gateway*** dashboard